Formal Methods for Hopfield-like networks

Hedi Ben Amor · Fabien Corblin · Eric Fanchon · Adrien Elena · Laurent Trilling · Jacques Demongeot · Nicolas Glade

Reviewed : 10 October 2012 / Accepted : 7 January 2013

Abstract Building a meaningful model of biological regulatory network is usually done by specifying the components (*e.g.* the genes) and their interactions, by guessing the values of parameters, by comparing the predicted behaviors to the observed ones, and by modifying in a trial-error process both architecture and parameters in order to reach an optimal fitness. We propose here a different approach to construct and analyze biological models avoiding the trial-error part, where structure and dynamics are represented as formal constraints. We apply the method to Hopfield-like networks, a formalism often used in both neural and regulatory networks modeling. The aim is to characterize automatically the set of all models consistent with all the available knowledge (about structure and behavior). The available knowledge is formalized into formal constraints. These last are compiled into Boolean formula in conjunctive normal form (CNF) and then submitted to a Boolean satisfiability solver. This approach allows to formulate a wide range of queries, expressed in

This work was supported by the European Network of Excellence 'Virtual Physiological Human' (VPH-NoE) through its PhD scholarship program, and by the Rhône-Alpes Complex System Insitute IXXI.

H. Ben Amor · N. Glade · A. Elena · J. Demongeot
AGIM Laboratory
Laboratory of AGeing Imaging and Modeling
University of Grenoble - CNRS FRE 3405
Domaine de la Merci, 38700 La Tronche, France
Tel. : +33-4-76-63-71-46
E-mail: Nicolas.Glade@ujf-grenoble.fr

F. Corblin · E. Fanchon · L. Trilling
TIMC-IMAG Laboratory
Laboratory of Techniques for biomedical engineering and complexity management - Informatics, Mathematics and Applications - Grenoble
University of Grenoble - CNRS UMR 5525
Domaine de la Merci, 38700 La Tronche, France
Tel. : +33-4-56-52-00-27
E-mail: Eric.Fanchon@imag.fr

a high level language, and possibly integrating formalized intuitions. In order to explore its potential, we use it to find cycles for 3-nodes networks and to determine the flower morphogenesis regulatory network of *Arabidopsis thaliana*. Applications of this technique are numerous and concern the building of models from data as well as the design of biological networks possessing specified behaviors.

Keywords Regulatory networks · Hopfield-like networks · Biological model building · Constraint-based programming · *Arabidopsis thaliana*

1 Introduction

Most biological processes imply regulatory relationships between proteins and genes at the cellular level, or between cells at the tissue level. Such systems are represented as interaction graphs composed of nodes representing the components of the system (genes, proteins, cells) linked together by directed arrows indicating the relationships between them.
However, the building of a model of biological regulatory network depends principally on two types of knowledge: structural and behavioral (or dynamical) knowledge. Structural knowledge (who are the actors, and who influences who?) can be extracted in a number of ways, *e.g.* via two-hybrid screening to identify protein-protein interactions or genetic experiments to find epistasis. Behavioral knowledge is directly inferred by observing patterns of expression of molecular markers in different cellular contexts. It corresponds to the overall genetic behavior (gene expression profile“) to which the cells of living organisms converge, and that generates their proteomes and subsequently their phenotypes. For example one can observe that gene g is expressed in cell type T_1 at a certain level whilst absent in cell type T_2. When designing a regulatory network, the choice of the modeler – which is correlated with the time and resources he can spend – will drive the needs in one type of knowledge, the other type or both types. Some models are preferentially based only on structural knowledge. One can of course simulate the behaviors of all the models describing a regulatory network and check which models reproduce the experimentally observed expression patterns. This is only possible when the number of models is small (reasonably below tens of billions models) as in (Giacomantonio and Goodhill 2010). In (Ben-Amor et al 2009), the authors analyze the fluorescence intensity of some genetic markers and infer the local structure of the 4 genes that induce the periodic spatial pattern needed for feather morphogenesis. Recently other authors (Gowda et al 2009) determined the structure of an interaction graph by measuring the correlation of gene expressions between consecutive time steps. The same idea of "directional correlation" had been previously proposed in (Demongeot et al 2003; Aracena et al 2003) and a pure logical inference method about the structure of the undirected version of the interactions graph had been described in (Aracena and Demongeot 2004). Other models are principally based on and emphasize the behaviors of the system given a certain network topology. The weights of the interactions are

adjusted to best fit a desired behavior. This adjustment is ensured by a learning process, as it is the case in soft computing when designing artificial neural networks.
The available data can also be a mix of structural and behavioral knowledge with different levels of abstraction, which can vary gradually from qualitative to quantitative levels. Mendoza and Alvarez-Buylla (1998) started with both types of knowledge when they modeled the biological network that regulates the flower morphogenesis of *Arabidopsis thaliana*. They used genetic algorithms to select a network having behaviors which fitted a desired one. A common strategy is to build up a tentative model of the system of interest (using only 'local' data, *i.e.*, binary interactions and values of kinetic parameters), then use the data of observed behaviors to compare them with predicted behaviors and validate or falsify the model. This strategy uses the available knowledge in two levels ; upstream and downstream the modeling process. Structural knowledge is integrated upstream and behaviors downstream. This considers that the behaviors are more reliable than the structural knowledge because this strategy converges toward the behavioral observations by modifying the structure of the network. Such consideration is a consequence of this decomposition and not a deliberate choice of the modeler. Moreover, the convergence is ensured through the maximization of a fitness function which may lead in some cases to a local minimum. One of our objectives here is to override this decomposition by considering all the knowledge and hypotheses at the same level of processing, where knowledge is no longer functionally divided but entirely integrated upstream. A formalization that considers initial data and hypotheses as constraints ensures such unification. Moreover, it renders the non-uniqueness of biological modeling by providing the complete set of consistent instantiations (solutions). In the context of modeling, the constraint-based approach entails a profound change of perspective: (i) As said above the relationship between structure and behaviors is not unidirectional anymore (from structure to behaviors when performing simulations/predictions). Structure and behaviors are both represented as constraints and exploited jointly by constraint solvers. This allows a much greater power of expression and flexibility in the type of questions which can be addressed. (ii) A set of constraints can have many solutions (under constrained problem), in which case there is no reason to single-out one solution.
This is in contrast with the traditional approach where a 'representative' solution is used, and from which predictions are made. Keeping in mind at all times that we are dealing with *solution sets* is an important change of mindset, and opens the way to the development of new functionalities. One can for example use the current knowledge to prioritize the next experiments to perform in order to reduce the set of solutions.
When using the constraint-based approach, a failure means that the hypotheses and the raw data are contradictory. In other words there is a contradiction between the assumed structure of the network and the observed or desired behaviors. It is important to realize that such result is obtained in one stroke without having to run numerous simulations in order to test all combinations

of bounded integer parameter values. The algorithm uses constraint propagation mechanisms which accelerate considerably the processing until getting a result (compared to the calculus over all possible simulations). In case of inconsistency one has to revise the model by putting into question some hypotheses defining the model. For example some interactions or genes have to be removed from the network (erroneous observations), or on the contrary new genes have to be added to the network (lack of information). If the pool of constraints is consistent it means that uncertain knowledge like intuitions and hypotheses can be kept in the pool. When more observations become available, new constraints are added to the pool, and consequently inconsistencies may appear, leading to a new phase of revision.

Whilst presenting some analogies, the present approach (reverse engineering) and model checking may not be confused. Model checking works indeed with only one model and does not imply any reverse engineering method like constraint propagation. Model checking is a technique that allows the automatic validation of discrete automata (mostly in computer sciences, electronics). It verifies if a given instantiated model (the system itself or an abstraction of it) satisfies a specification often formulated in terms of temporal logic (e.g. CTL). For example, it can be used successfully to find the number of attractors in some well defined boolean network based models of regulatory network, as described by Dubrova and Teslenko (2011). We do not address the same problem. Ours is larger: given a parameterized family of models, we want to find the solution set of instantiated models (models in which the parameters are defined) that all satisfy a given interaction graph and a specification. We obtain by this way the set of existing and satisfying models given a necessary structure (common to all models of the solution set) and behaviour of the network, both knowledges expressed in a language like CTL but more expressive. We also aim to automatically revise the initial model if it presents any incoherence (no solutions obtained). Our approach allows to infer properties (initially not known) which are common to all coherent models (see for example the study of the effects of signs in Section 4.1). The study of the instantiated models in the solution set, if not too large, allows to extract conclusions on one or several models of network satisfying the knowledge on the structure and behaviour.

The main work done here concerns the formalization of a variant of Hopfield networks (Hopfield 1982), which we call here *Hopfield-like networks*, in the form of constraints on integers. Boolean automata networks are among the most used models in biological modeling of regulatory networks. Initially, they were introduced by Kauffman to study global properties of genetic nets (Kauffman 1969). To model a particular regulatory phenomenon, we chose a – Boolean – thresholded automaton. This model is similar to Hopfield's model (Hopfield 1982) but it is discrete and more general in the sense that there are no conditions of symmetry imposed on the weights and self-interaction loops are authorized (as illustrated in Figure 1). Two other notable advantages of Hopfield-like networks are their intuitive notation for biologists and the possibility to take into account different update schedules (parallel, sequential, block-parallel). In the present paper, we only focus on parallel update sched-

ules but constraints can also be written to take into account the different update schedules. Two applications are presented in the *Results* section to illustrate the feasibility and the potentiality of this approach: the first one is inspired from theoretical questions, the second one is a biological inference problem. We demonstrate here the capacity of this method to help biologists to design consistent regulatory networks given a certain knowledge about the system of interest.

2 Formalization of Hopfield-like networks

A Hopfield network (as a Hopfield-like network) H is composed of nodes (e.g. genes) g_i, $i \in 1..n$, associated to thresholds θ_i, and of oriented edges from node g_j to g_i associated to weights w_{ij}. The vector of thresholds is noted θ_H, and the matrix of weights is noted W_H (Figure 1). In classical Hopfield networks, self-interactions (edges from g_i to g_i) are forbidden, the matrix of weights is symmetric, and the parameters θ_i and w_{ij} take real values. In our case of study, **Hopfield-like networks**, no restriction is made on the network topology and the parameters θ_i and w_{ij} take signed integer values limited to an interval $[-Max..Max]$. We represent a Hopfield network H by the couple (W_H, θ_H).

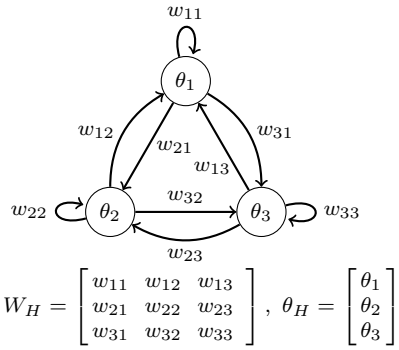

Fig. 1 Example of Hopfield-like network H fully connected with 3 nodes.

A state S of H is a vector $\langle S_1, S_2, ..., S_n \rangle$ where $S_i \in \{0, 1\}$ is the value of the node g_i in S. The behaviors of H are ruled by a state transition graph containing 2^n network states S^k. In the following, node indices will be noted as subscripts whereas network state indices as superscripts (*e.g.* S_i^k *is the value of* g_i *in state* S^k). The existence of a transition $S^k \nrightarrow S^{k'}$ between the network states, noted $transition(H, S^k, S^{k'})$, is defined by Definition 1.

Definition 1 : $transition(H, S^k, S^{k'}) \Leftrightarrow \bigwedge_i \ [S_i^{k'} \Leftrightarrow [(\sum_j w_{ij}.S_j^k) > \theta_i]]$

$$where \;\; W_H = \begin{bmatrix} . & . & . \\ . & w_{ij} & . \\ . & . & . \end{bmatrix} \;\; , \;\; \theta_H = \begin{bmatrix} . \\ \theta_i \\ . \end{bmatrix}$$

Note that, each state S^k has a unique successor $S^{k'}$. The component $S_i^{k'}$ which represents the value of g_i in $S^{k'}$ is determined by the truth value of the inequality $[(\sum_j w_{ij}.S_j^k) > \theta_i])$. This is the formalization of Hopfield-like network synchronous updating: the next state of a node g_i is computed by comparing the sum of effective weights to its threshold θ_i.

Example 1 *We consider the example of Figure 1 with the following valuation of parameters:*

$$W_H = \begin{bmatrix} -1 & -2 & -1 \\ 1 & -1 & 2 \\ -2 & 2 & 3 \end{bmatrix} \;\; , \;\; \theta_H = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}$$

By applying the Definition 1 for the state $S = 110$, we obtain its successor state $S' = 000$ of S with $S'_1 \Leftrightarrow -1 - 2 > -3$, $S'_2 \Leftrightarrow 1 - 1 > 0$, and $S'_3 \Leftrightarrow -2 + 2 > 0$.

We give in Figure 2 all the transitions: depending on its initial state, the system reaches a fixed point or a cycle. We note that each of those two has its own attraction basin, i.e. the set of initial states leading to it.

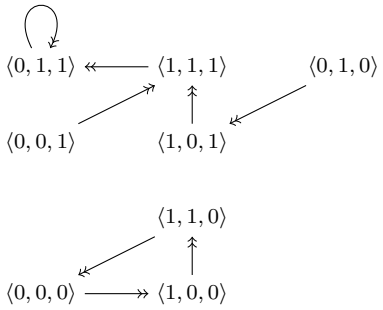

Fig. 2 Graph of transitions for the Example 1. Note the two attractors of the network: one fixed point $\langle 0, 1, 1\rangle$ and one cycle $[\langle 0, 0, 0\rangle, \langle 1, 0, 0\rangle, \langle 1, 1, 0\rangle]$.

We also introduce a relation $path(H, p, q)$ which is true if the list of q states P is a path in H.

Definition 2 *:* $path(H, P, q) \Leftrightarrow \bigwedge_{k \in 1..q-1} transition(H, P^k, P^{k+1})$
where P^k is the k^{th} element of P

Definition 1 defines the formal relation between the existence of a transition $S^k \twoheadrightarrow S^{k'}$ and the variables defining the network structure θ_i's and w_{ij}'s. It is the building block to express queries involving paths and attractors. The aim of the game is thus to find a way to express a given problem in terms of constraints using the relation $path/3$ (3 is for the arity, number of parameters, of the relation). The constraint set is then submitted to constraint solvers which implements deduction rules (constraint propagation in the jargon) and enumeration strategies.

3 Implementation of Hopfield-like networks

The implementation uses the environment SICStus Prolog, where the constraints, directly over bounded integer variables, are encoded via the library CLP(FD) (Constraint Logic Programming (Finite Domain)) (Carlsson et al 1997; Apt 2003).

The form of Definition 1 is actually not suitable for efficient constraint propagation. We present in Section 3.1 its translation to a more suitable form for finite domain solvers.

Moreover, we can use the finite domain solver in cooperation with a SAT solver, depending on the complexity of the queries. The SAT solver used is MiniSAT (Eén and Sörensson 2004; Eén and Biere 2005). This second type of solvers is extremely efficient at computing the satisfiability of very large formulas in Conjunctive Normal Form (CNF) which is the standard format for the classical Boolean satisfiability problem (SAT).

We present some necessary relations in Section 3.2 to permit an easy translation into CNF formulas.

3.1 Translation for finite domain solver

In order to obtain, from Definition 1, a formalization directly suitable for finite domain solvers, we need to introduce some intermediate variables with finite domains. The Proposition 1, using these intermediate variables, gives the mean to implement into constraints the relation $transition/3$ for finite domain solvers.

Let g_i be one of the network nodes, and let L_i be the subset of nodes g_j that have an influence on g_i (in other words there is an arc from g_j to g_i). Let us note $|L_i|$ the cardinal of L_i. From the viewpoint of node g_i, there are $2^{|L_i|}$ possible *contexts*, depending on the state of the $|L_i|$ influencing nodes: The contexts are the equivalence classes of network states according the equivalence "The active predessors of g_i are the same". The contexts are a set of subsets of network states.

To represent the context of each node g_i, we introduce the notion of neighboring state l_i, an index made of $|L_i|$ binary digits, which is a sequence of

state values of the influencing nodes. The order of the digits in l_i follows the numbering of the nodes. We denote L_{i,l_i} the subset of L_i containing the nodes whose state value is equal to 1 in l_i. Each l_i defines a unique set L_{i,l_i}. We call $Context_{i,l_i}$ the context defined by l_i, that is the set network states whose values are in accordance with l_i.

Definition 1 contains the expression $\sum_{j\in L_i} w_{ij}.S_j^k$, corresponding to at most $2^{|L_i|}$ possible sums $\sum_j w_{ij}$, depending on which nodes $j \in L_i$ are active ($S_j^k = 1$). To represent the summation $\sum_{j\in L_i} w_{ij}.S_j^k$ we introduce $2^{|L_i|}$ independent variables $\mathbf{Sum_{i,l_i}} = \sum_{j\in L_{i,l_i}} w_{ij}$

Example 2 *Let us consider the example given in Figure 1, but where the arc from node* 2 *to* 1 *is suppressed. Then* $L_1 = \{1, 3\}$*. So, there are* $2^2 = 4$ *possible subsets* L_{1,l_1}*, and consequently* 4 *new variables. For example:*
- with $l_1 = 11$*, we have* $L_{1,11} = \{1, 3\}$*,* $Context_{1,11} = \{101, 111\}$ *and* $Sum_{1,11} = w_{11} + w_{13}$*,*
- with $l_1 = 01$*, we have* $L_{1,01} = \{3\}$*,* $Context_{1,01} = \{001, 011\}$ *and* $Sum_{1,01} = w_{13}$*.*

We introduce a second type of variables $\mathbf{InContext^k_{i,l_i}}$ with Boolean domains, which are true when state S^k belongs to $Context_{i,l_i}$. These variables are such that $\mathbf{InContext^k_{i,l_i}} = (\bigwedge_{j\in L_{i,l_i}} S_j^k) \wedge (\bigwedge_{j'\in L_i\setminus L_{i,l_i}} \neg S_{j'}^k)$

Given a node g_i, a state S^k belongs to one and only one context. For each state S^k, there are $2^{|L_i|}$ $InContext^k_{i,l_i}$ variables, *i.e.* one for each context of g_i. The union of the $2^{|L_i|}$ contexts $Context_{i,l_i}$ associated to a node g_i is equal to the whole state space. When a question contains several formal states $S^1, ..., S^k, ..., S^{k'}, ...$, as it is the case usually (see below), the network state index k specifies which state the $InContext^k_{i,l_i}$ variable is defined from.

Proposition 1 *:*
$transition(H, S^k, S^{k'}) \Leftrightarrow \bigwedge_i \bigwedge_{l_i} [InContext^k_{i,l_i} \Rightarrow [S_i^{k'} \Leftrightarrow (Sum_{i,l_i} > \theta_i)]]$

Let us note by (1) the formula $\bigwedge_i [S_i^{k'} \Leftrightarrow [(\sum_j w_{ij}.S_j^k) > \theta_i]]$ of Definition 1 and by (2) the formula $\bigwedge_i \bigwedge_{l_i} [InContext^k_{i,l_i} \Rightarrow [S_i^{k'} \Leftrightarrow (Sum_{i,l_i} > \theta_i)]]$ of Proposition 1. (1) and (2) are equivalent.
In fact, let g_i a component. Let S^k a state. Any state belongs to one and only one context. Let l_i the context for S_k. So $InContext^k_{i,l_i} = 1$, and for any $l_{i'} \neq l_i$ we have $InContext^k_{i,l_{i'}} = 0$. Due to the definition of $InContext^k_{i,l_i}$, we obtain $Sum_{i,l_i} = \sum_{j\in L_{i,l_i}} w_{ij} = \sum_j w_{ij}.S_j^k$. This equality is noted (3).
Suppose that (1) is true. Then $S_i^{k'} \Leftrightarrow (Sum_{i,l_i} > \theta_i)$ (using equality (3)). As this equivalence is the sole to be true among the choice of contexts in (2) (for any $l_{i'} \neq l_i$ we have $InContext^k_{i,l_{i'}} = 0$), then (2) is true.
Suppose that (2) is true. As $InContext^k_{i,l_i} = 1$ and for any $l_{i'} \neq l_i$ we have $InContext^k_{i,l_{i'}} = 0$, then $S_i^{k'} \Leftrightarrow (\sum_j w_{ij}.S_j^k > \theta_i)$ (using equality (3)). So (1) is true.

3.2 Translation for SAT solver

In (Corblin et al 2010, 2011), we present our translation into CNF of several types of constraints over unsigned integer and Booleans (in particular reified constraints as $B \Leftrightarrow X > Y$). The idea, to fill the gap between expression in Proposition 1 and a CNF expression, is to reify constraints and treat only unsigned integer variables.

The expression in Proposition 1 contains constraints having arity superior or equal to 3, but they can be decomposed into binary or ternary constraints by a process called reification. For example, the 4-arity constraint $(X < Y \Leftarrow Z < Y)$ is equivalent to $(B1 \Leftrightarrow X < Y) \wedge (B2 \Leftrightarrow Z < Y) \wedge (B1 \Leftarrow B2)$, a conjunction of binary and ternary constraints involving two additional Boolean variables.

In order to express our problem with unsigned integer variables only, we represent signed integer variables, Sum_{i,l_i} and θ_i, by couples (σ, V), where σ is a Boolean which is true if and only if the represented signed integer is positive or null, and V is the absolute value of the represented signed integer.

In addition, two new relations have to be defined: the one defined in Definition 3, to formalize the addition of two signed integers X and Y (used to translate into CNF Sum_{i,l_i}, and the other defined in Definition 5, to translate into CNF "B equivalent to $X > Y$" with X and Y two signed integers, which is necessary to translate into CNF $Sum_{i,l_i} > \theta_i$.

Definition 3 *:*

$$
\begin{aligned}
c\_sgn\_AddXYZ(X,Y,XpY) \Leftrightarrow \quad & X = (\sigma_X, V_X) \wedge \\
& Y = (\sigma_Y, V_Y) \wedge \\
& XpY = (\sigma_{XpY}, V_{XpY}) \wedge \\
& B_{le} \Leftrightarrow V_X \leq V_Y \wedge \\
& B_{ge} \Leftrightarrow V_X \geq V_Y \wedge \\
& minmax\_B\_le(V_X, V_Y, B_{le}, Min_{V_X,V_Y}, Max_{V_X,V_Y}) \wedge \\
& V1_{XpY} = V_X + V_Y \wedge \\
& V2_{XpY} = Max_{V_X,V_Y} - Min_{V_X,V_Y} \wedge \\
& (\sigma_X \Leftrightarrow \sigma_Y) \Rightarrow (V_{XpY} = V1_{XpY}) \wedge \\
& (\sigma_X \not\Leftrightarrow \sigma_Y) \Rightarrow (V_{XpY} = V2_{XpY}) \wedge \\
& \sigma_{XpY} \Leftrightarrow (\sigma_X \wedge \sigma_Y) \vee (\sigma_X \wedge B_{ge}) \vee (\sigma_Y \wedge B_{le})
\end{aligned}
$$

The relation $c\_sgn\_AddXYZ(X,Y,XpY)$ is true if and only if X, Y and XpY are signed integer variables represented by couples such as (σ, V), and XpY is equal to $X + Y$.

Definition 4 *:*

$$
\begin{aligned}
minmax\_B\_le(V_X, V_Y, B_{le}, Min, Max) \Leftrightarrow \quad & (B_{le} \wedge Min = V_X \wedge Max = V_Y) \vee \\
& (\neg\, B_{le} \wedge Min = V_Y \wedge Max = V_X)
\end{aligned}
$$

The relation $minmax\_B\_le(V_X, V_Y, B_{le}, Min, Max)$ is true if and only if Min (resp. Max) is the minimum (resp. maximum) of V_X and V_Y.

Definition 5 *:*

$$\begin{aligned} c\_sgn\_SupXYB(X,Y,b) \Leftrightarrow \quad & X = (\sigma_X, V_X) \wedge \\ & Y = (\sigma_Y, V_Y) \wedge \\ & b \Leftrightarrow \\ & (\sigma_X \wedge \neg\, \sigma_Y) \vee \\ & (\sigma_X \wedge V_X > V_Y) \vee \\ & (\neg\, \sigma_Y \wedge V_X < V_Y) \end{aligned}$$

The relation $c\_sgn\_SupXYB(X, Y, B)$ is true if and only if X and Y are signed integer variables represented by couples such as (σ, V), and B is a Boolean equivalent to $X > Y$.

4 Results

In this section, two examples of questions are described. All concern Hopfield-like networks having a parallel update schedule. Different update schedules like sequential or block-parallel can be taken into account by adding new constraints. For all the queries presented below, the set of all consistent instantiations (solutions) is obtained.

4.1 Finding cycles for 3-nodes networks

A typical behavior of a network one could obtain is the existence of a – limit – cycle of a given length p, *i.e.* having a given number of transitions. Definition 6 defines the relation $cycle(H, C, p)$ true if and only if C is a cycle of length p produced by the network H.

Definition 6 *:*

$cycle(H, C, p) \Leftrightarrow$
$C = [S^1, S^2, ..., S^{p+1}] \wedge path(H, C, p) \wedge all\_diff([S^1, ..., S^p]) \wedge S^1 = S^{p+1}$

where $all\_diff(L)$ is true if and only if all the states of the list L are all differents.

Networks satisfying the relation $cycle/3$ defined in Definition 6 exist. An example of solution for a 3-node network and in the case of $p = 2^n$ (the length of the cycle is equal to the number of possibles states, *i.e.* $p = 2^3 = 8$) is given in Figure 3.

$$W = \begin{bmatrix} -1 & 1 & 1 \\ -1 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix}, \ \theta = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

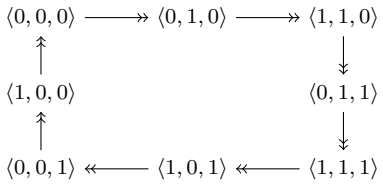

Fig. 3 Example of a network having all its possible states in a single cycle of size 8.

In Question 1 we ask if there is a cycle of length p for a network with only positive values for the network parameters $w_{i,j}$ and θ_i.

Question 1 *: Is there a Hopfield-like networks $H = (W, \theta)$ coherent with the formula $positive(W) \wedge positive(\theta) \wedge cycle(H, C, p)$?*

where $positive(M)$ is true if and only if M is a matrix of positive or null integers.

The answer of Question 1 is "yes". In fact, there are several solutions (in terms of parameter values). One is given in the low part of Figure 4.

	$\theta_j < 0$	$\theta_j \geq 0$	$\forall\ \theta_j$
$W_{ij} \leq 0$	2,3,6,8	No cycle	2,3,6,8
$W_{ij} \geq 0$	No cycle	2 ,3	2,3
$\forall\ W_{ij}$	2..8	2,3,4,5,6,8	2..8

$$W = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

$$\theta = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

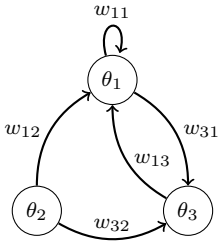

Fig. 4 Lengths of existing cycles depending on the signs of weights and thresholds for a 3-node network. The system, in the low part, corresponds to the case $W_{ij} \geq 0$ and $\theta_j \geq 0$ (highlighted in the table) and has one cycle of length 2 (between states 100 and 001).

Now, we explore deeply the effect of the sign of parameters over the length of cycles by adpating the Question 1 (modification of p and the constraints

on the signs). The results on the length of existing cycles as a function of the sign of weights and thresholds are given in the top part of Figure 4.

Given a network of size 3, no cycles are obtained when the weights are positive and the thresholds negative or when the weight are negative and the thresholds positive.

R. Thomas conjectured (Thomas 1980) that a negative circuit is a necessary condition for stable periodicity. This conjecture has been formally proven in the context of discrete networks (Remy et al 2008; Richard 2010). In the case of Remy et al (2008) the strategy can be deduces from the attractive cycle, and in the case of Richard (2010) the considered dynamics is the asynchronous one which is undeterministic.

The example given in the low part of Figure 4 shows that it is not verified in the case of a parallel update strategy.

In case of unsatisfiability (no cycle), the constraints on the signs must be removed to find cycles. This answer allows us to make a bridge between structure and behaviors in the whole set of networks.

4.2 Defining the regulatory network of Arabidopsis thaliana morphogenesis

Finding models of regulatory networks can be easily and very efficiently done via constraints (Corblin et al 2009). In (Mendoza and Alvarez-Buylla 1998), the authors have designed a model, noted here H_A, of the morphogenesis of the *Arabidopsis thaliana* flower by using a genetic algorithm on a population of networks. They kept the solution that – in this context of genetic algorithm – best fitted the experimental observations and was consistent with an existing model called the ABC model (Coen and Meyerowitz 1991). The parameters, obtained by Mendoza and Alvarez-Buylla (1998), and the behaviors of H_A, which we have computed algorithmically in the case of a parallel update strategy, are shown in Figure 5. Here, we aim to obtain the set of similar networks having at least the behaviors described in (Mendoza and Alvarez-Buylla 1998) by using constraint–based methods.

We synthesize in the form of constraints the whole knowledge that these authors used. They started with structural knowledge which consists of inequalities between the weights of gene interactions involved in the regulation of the flower morphogenesis. Then they checked the obtained simulated behaviors with behavioral knowledge from the ABC Model (Coen and Meyerowitz 1991). The ABC model postulates that 3 types of activities specify the different organs of the flower; activity A specifies sepals, coupled activities A and B specify petals, coupled activities B and C specify stamens and finally activity C specifies carpels. In the *Arabidopsis* case, A corresponds to the expression of the gene *AP1* (4^{th} node), B to the joint expression of *AP3* and *PI* (10^{th} and 11^{th} nodes) and C to the expression of AG (9^{th} node). In addition to this knowledge, the authors introduced the graph of existing interactions (with non null weights). We give in Figure 6 this network. Moreover, we note

$$W_A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & -2 & 1 & -2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & -2 \\ 0 & 0 & 4 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \theta_A = \begin{bmatrix} 0 \\ 0 \\ 3 \\ -1 \\ 1 \\ 0 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Behaviors of H_A :

#	Attractor	Cell Type	SAB
F1	000100000000	Sepal	168
F2	000000001000	Carpel	24
F3	000100010110	Petal	248
F4	000000011110	Stamen	8
F5	110000010110	Mutant	384
F6	110000000000	No Flower	384
C1	000000000000 000100001000	—	192
C2	000100001110 000000010000	—	272
C3	110000000110 110000010000	—	1280
C4	000100000110 000100010000	—	800
C5	000000001110 000000011000	—	32
C6	000000000110 000100011000	—	176
C7	000100011110 000000010110	—	128

Fig. 5 Parameters of the network H_A and its behaviors. H_A is the network obtained by Mendoza and Alvarez-Buylla (1998). The column SAB contains the size of the attraction basins.

structure_mendoza(**H**) the relation which defines the Hopfield-like network with 12 nodes of this figure with the notations of weights used originaly by the authors Mendoza and Alvarez-Buylla (1998) (for example a for the weight of edge from node 3 onto node 4).

It has to be said that many constraints introduced by Mendoza and Alvarez-Buylla (1998) are not meaningful. These authors indeed compared weights that should not be compared because they do not belong to interactions over the same target gene. For example, weights a and b correspond to interactions that do not concern the same target: interaction a acts on node 4 while b acts on node 3. On the contrary weights e and n can be compared because both act on the same node 11. However, we understand what these authors wanted to express: probably as many biologists, they were tempted to describe in the form of weights the relative strengths of interactions between genes acting on

different targets, because this make sense in biological terms. A gene g_1 can be much more sensitive to the action of the product of another one g_2, than another third gene g_3 would be for the products of a fourth one g_4, and this because of different levels of expression of genes g_2 and g_4, and because of the different efficiencies of the promoters of g_1 and g_3.

To be in agreement with Hopfield's formalism, we only kept the weight comparisons that are permitted (see Constraint 1), *i.e.* those that are in the same line in the interaction matrix.

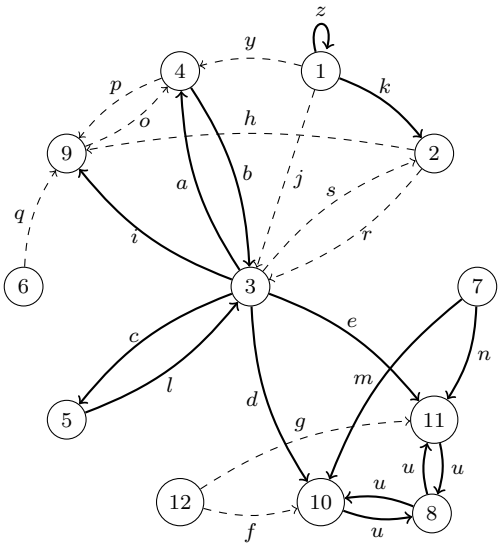

Fig. 6 The graph of interactions of the *Arabidopsis thaliana* flower morphogenesis regulatory network (*cf.* Mendoza and Alvarez-Buylla (1998)). The solid (resp. dashed) edges are activations (resp. inhibitions). The letters on arcs identify the weights (these variables are used in Definition 1). According to the authors, the genes from 1 to 12 correspond, in the same order, to *EMF1*, *TFL1*, *LFY*, *AP1*, *CAL*, *LUG*, *UFO*, *BFU* (the *AND* function), *AG*, *AP3*, *PI* and *SUP*.

We define below : the structural knowledge in Constraint 1, the activities associated to the ABC model in Constraint 2 and the behavioral knowledge in Constraint 3

Constraint 1 *:*

$$
\begin{aligned}
structure(H) \Leftrightarrow\ & structure\_mendoza(H) \wedge \\
& a > |o| \wedge \\
& b > l \wedge \\
& d > |f| \wedge \\
& e > n \wedge e > |g| \wedge \\
& |h| > i \wedge \\
& |p| > |q| \wedge \\
& u = 1 \wedge \\
& z = 1 \wedge \\
& \theta_1 = 0 \wedge \\
& \theta_6 = 0 \wedge \\
& \theta_7 = 0 \wedge \\
& \theta_{12} = 0 \wedge \\
& \theta_8 = 1
\end{aligned}
$$

In this definition we express the structural knowledge used by the authors and we suppose that the thresholds of the sources are null. The sources are the non-regulated genes (nodes 6,7,12) and the self-regulated gene (node 1). Their threshold values are set to zero ($\theta_i = 0$ for $i \in \{6, 7, 12, 1\}$) to avoid a permanent expression because they are not regulated by other genes in the network. The case of a permanent expression of the sources (boundaries) is interesting but it corresponds to a robustness study towards noise and external factors (Ben-Amor et al 2008, 2009). Therefore, it is out of our scope here.

The 8^{th} node, stands for *BFU* (*Boolean Function*), was introduced in (Mendoza and Alvarez-Buylla 1998) to represent the protein heterodimer formed by *AP3* (10^{th} node) and *PI* (11^{th} node). This complex forms an active transcription factor. The authors formalized this by an *AND* logical function acting back on *AP3* and *PI*. The weights and parameters of this hypothetical pattern in the network are well defined. This choice is made in such a way that we can express the complex formation in a Hopfield model. All the weights of the interactions in this pattern are equal to 1 and the threshold of activation of *BFU* is equal to 1. The introduction of this node implies the function B (as defined above) to be the expression of *BFU* or (inclusively) the joint expression of *PI* and *AP3*.

Note that some other inequalities are given by Mendoza and Alvarez-Buylla (1998). They are given by the following formula: $a > b \wedge a > c \wedge a > d \wedge a > e \wedge c > l \wedge d > m \wedge d > n \wedge e > d \wedge |f| > |g| \wedge |j| > k \wedge m > n \wedge |p| > |q| \wedge |s| > |r|$. They involve comparisons between interaction weights acting on differents nodes. In this formalism (Hopfield-like network), the weight represents the contribution of an entity to the activation of another. This should not be confused with quantitative modeling. For this reason we omit them. Let's call them supplementary inequalities.

Constraint 2 *:*

$abc\_function_A(S) \Leftrightarrow S_4$
$abc\_function_B(S) \Leftrightarrow S_{10} \wedge S_{11}$
$abc\_function_C(S) \Leftrightarrow S_9$

These 3 constraints formalize the Boolean functions used in the ABC model (see above).

Constraint 3 *:*

$$
\begin{aligned}
dynamic(H) \Leftrightarrow\ & transition(H, S^1, S^1) \wedge && \%\ Steady\ \ states \\
& transition(H, S^2, S^2) \wedge \\
& transition(H, S^3, S^3) \wedge \\
& transition(H, S^4, S^4) \wedge \\
& abc\_function_A(S^1) \wedge && \%\ Sepal\ \ Attractor \\
& \neg\ abc\_function_B(S^1) \wedge \\
& \neg\ abc\_function_C(S^1) \wedge \\
& abc\_function_A(S^2) \wedge && \%\ Petal\ \ Attractor \\
& abc\_function_B(S^2) \wedge \\
& \neg\ abc\_function_C(S^2) \wedge \\
& \neg\ abc\_function_A(S^3) \wedge && \%\ Stamen\ \ Attractor \\
& abc\_function_B(S^3) \wedge \\
& abc\_function_C(S^3) \wedge \\
& \neg\ abc\_function_A(S^4) \wedge && \%\ Carpel\ \ Attractor \\
& \neg\ abc\_function_B(S^4) \wedge \\
& abc\_function_C(S^4)
\end{aligned}
$$

In this constraint we express the behavioral knowledge used by the authors. The constraint on the dynamics asks for *at least* 4 fixed points corresponding to the 4 floral tissues, described by the ABC model : carpel, stamen, sepals and petals. In Constraint 3 we chose to keep the possibility of additional fixed points and cycles. It allows notably to find other 'tissues' like a mutant 'no flower' as shown in Fig. 8. Of course, constraints on the behavior can be added so that the behavior contains only 4 fixed points or no cycles (*NB.* we did not find any existing solutions without cycles).

Question 2 *: Is there at least one solution to the set of integrated data of Constraints 1 and 3 ?*

The answer of Question 2 is "yes". An example of solution, noted H_S, is given in the top part of Figure 8.

In fact, there are a lot of solutions (in terms of parameter values). Given a certain range of parameter values (for example, weights can be chosen in the interval $[-10, 10]$ or in a smaller interval like $[-2, 2]$) the set of models can be huge but a lot of solutions are equivalent. In fact, different values of parameters (weights and thresholds) give the same transition graph.

Two models are equivalent when they have the same behavior (set of transitions). This brings us to define the notion of 'minimal model' in thresholded integer automata networks. A model M defined by a vector of thresholds θ and a matrix of weights W, is called minimal when there is no equivalent model M' defined by a vector of thresholds θ' and a matrix of weights W' and such that $\exists\ i\ /\ |\theta'_i| < |\theta_i|$ or $|\sum_j w'_{ij}| < |\sum_j w_{ij}|$. An example is shown in Figure 7 and discussed in Example 3.

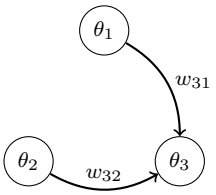

Fig. 7 The interaction graph of an AND Boolean function.

Example 3 *Let us consider the following values of parameters* $w_{31} = w_{32} = 1$, $\theta_1 = \theta_2 = 0$ *and* $\theta_3 = 1$ *for the network of Figure 7. These values give us an AND Boolean function as well as the following ones* $w_{31} = w_{32} = 2$, $\theta_1 = \theta_2 = 0$ *and* $\theta_3 = 2$. *One could not obtain another model of the AND Boolean function with integer values of parameters smaller than the first ones.*

For each model in the list of solutions, the algorithm attemps to reduce the values of parameters without changing the dynamics. If this is possible the model is eliminated because the solution with reduced parameter values is also in the list. This algorithm is detailed in (Elena 2009; Glade et al 2011)

If we consider the structural knowledge (Defintion 1) but not the behavioral knowledge (Defintion 3), and if we restrict the domains of w_{ij} and θ_i to adequate intervals (for example, for any function having 3 arguments, the weights are in the interval $[-2; 2]$ and the thresholds $[-3; 2]$), we obtain more than 37.3 10^{10} models (including equivalent ones).

If we add the dynamical knowledge (Constraint 3), there were 3360 models (including equivalent ones). By applying the algorithm of model minimization described before, we ensured that no equivalent models remained. We obtain, from the previous 3360 models, a set of 532 models (without equivalent ones).

All models have behaviors which converge to the experimentally observed stationary points or some showed other stationary points or showed different cycles. Mendoza's model does not belong to them because of the presence in this model of unsatisfiable constraints. In fact, Mendoza added more inequalities (called supplementary inequalities) which are not relevant with the formalism of thresholded automata networks. When one add these inequalities as new constraints, this leads to an unsatisfiability. One can not find weight values in $[-2, 2]$ and threshold values in $[-3, 2]$ satisfying all the inequalities imposed by Mendoza and expect to have the ABC stationnary points.

The modeler can then add additional criteria if for some reason fewer or only one model must be chosen. For example one can decide to take as criterion a score R defined as a function of the size of the basins of attraction of the different tissues.

This criterion could be used as a measure of structural robustness (*e.g.* under perturbations the larger the attraction basins the more stable it is in general) (Glade et al 2011).

5 Conclusions

We showed in this article how various types of queries can be implemented as sets of constraints: the initial knowledge is written naturally by using this paradigm. Then, we are able to impose constraints concerning the behaviors and the structure so as to get the set of models consistent with the biological knowledge the biologists will increase by their observations and hypothesis until reaching the uniqueness of the model.

In the real life, the initial knowledge and intuitions are incomplete, so we will have more than one solution, although it is also possible to have a contradiction, in which case there is no solution at all. In this investigation, we used two sets of constraints. The first one contains all the constraints originally used by Mendoza. These constraints lead to a contradiction. The second set of constraints is obtained by removing the supplementary inequalities (see above). This second set is satisfiable and give us 532 possible instantiations of parameters. By showing that there is no unique model of *Arabidopsis thaliana* flower morphogenesis' regulatory network, and that other possible behaviors exist (additional cyclic attractors that could correspond to a rhythmic activity in the cell), we aimed to warn biologists against a too-confident design of their own models. Biologists often construct their models by following a trial-error approach, that progressively converge to one of the plausible models which sometimes becomes a reference in the literature. We point out that most of the time their models are not unique. More, our technique constitutes a powerful tool for inferring new properties (not thought before). These properties concern essentially the interactions (existence, manner to compose interactions) and conditions about one specific state of a path. We will work soon over more general properties about dynamics (cycles synchronization, description of bassins of attraction). Moreover, the idea is to go toward experiment design. With our approach, the idea is to infer scoring properties about perturbation and observation of the system. We have applied this approach to the modeling of the nutritional stress network of *E. coli* using the formalism of R. Thomas (Corblin et al 2009). Here we used, in this context of inverse methods, the formalism of Hopfield-like networks, which is a novelty in itself and enlarges the field of applications. We aim now to adapt our technique to more general logical formalisms that include thresholded automata but also other simple (*AND*, *OR*, *XOR*, ...) or complex (modules composed of several logical

$$W_{H_S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & -3 & 1 & -2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 1 & 3 & 0 & 0 & 0 & -1 \\ 0 & 0 & 2 & 0 & 0 & 0 & 1 & 3 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \theta_{H_S} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ -1 \\ -1 \\ 1 \\ -2 \\ 3 \\ 3 \\ 0 \end{bmatrix}$$

Behaviors of H_S :

#	Attractor	Cell Type	SAB
F1	001111100000	Sepal	630
F2	000001101000	Carpel	42
F3	001111110110	Petal	126
F4	000001111110	Stamen	2
F5	110001100000	NoFlower	252
F6	101111100000	—	240
F7	101111110110	—	80
F8	110001110110	—	12
C1	001001100000 000111101000	—	384
C2	111001100000 100111100000	—	564
C3	000111101110 001001110000	—	152
C4	111001110000 100111100110	—	284
C5	001111100110 001111110000	—	588
C6	101111110000 101111100110	—	320
C7	110001110000 110001100110	—	120
C8	111001100110 100111110000	—	108
C9	000001101110 000001111000	—	20
C10	001001100110 000111111000	—	72
C11	000111111110 001001110110	—	32
C12	111001110110 100111110110	—	68

Fig. 8 Parameters of the network H_S and its behaviors. H_S is one selected instantiation (based on the maximization of a robustness criterion R from the set of consistent solutions we have obtained. R is the sum of the size of the attraction basins, in column SAB, of the plant tissues minus the one of the 'no flower' attractor).

functions) logical-based functions. The set of solutions may be huge, in which case it is not possible nor useful to enumerate the solutions. In the case of the

Arabidopsis thaliana flower morphogenesis regulatory network it was possible to perform the enumeration and one can apply an additional optimization criterion. As pointed out by Alon (2003), modularity and the use of recurring circuit elements are structural principles shared by biological and engineered networks. We can take advantage of these principles to reduce the number of models or to design new Hopfield-like networks by using known recurring modules as additional constraints. In addition, many models are very similar and will not be differentiated unless having very precise biological data. We will now develop criteria of classification and taxonomies by considering the probabilities of transition between network states and the size of the basins of attraction (robustness criteria), in order to extract the best instantiation (the most centered) or showing the existence of classes of models having similar structures. Some of them may be more similar than others like individuals within species.

Finally, we focused on parallel updated networks, but more interesting and biologically adapted update schedules that are not parallel (synchronous) or sequential (asynchronous), but block-parallel or block-sequential have to be considered now (Demongeot et al 2008). Of course, an effort has to be made now to propose a set of graphical tools or high level languages to convert biological data in an appropriated notation for existing SAT solvers, eventually by the way of automatic annotations in electronic laboratory notebooks.

Acknowledgements The authors are very grateful to O. Bastien (PCV, CEA Grenoble, France) and J. Cohen (Brandeis University, USA) for their help and stimulating discussions.

References

Alon U (2003) Biological networks: The tinkerer as an engineer. Science 301:1866–1867

Apt K (2003) Principles of Constraint Programming. Cambridge University Press

Aracena J, Demongeot J (2004) Mathematical methods for inferring regulatory networks interactions: Application to genetic regulation. Acta Biotheoretica 52:391–400

Aracena J, Lamine SB, Mermet M, Cohen O, Demongeot J (2003) Mathematical modelling in genetic networks: relationships between the genetic expression and both chromosomic breakage and positive circuits. IEEE Trans Systems Man Cyber 33:825–834

Ben-Amor H, Demongeot J, Sené S (2008) Structural sensitivity of neural and genetic networks. In: Springer (ed) LNCS 5317 Proceedings of 7th Mexican International Conference on Artificial Intelligence, 2008 (MICAI'08), pp 973–986

Ben-Amor H, Cadau S, Elena A, Dhouailly D, Demongeot J (2009) Regulatory networks analysis: Robustness in biological regulatory networks. In: IEEE

(ed) IEEE Proceedings of International Conference on Advanced Information Networking and Applications Workshops, 2009 (AINA'09), pp 924–929
Carlsson M, Ottosson G, Carlson B (1997) An open-ended finite domain constraint solver. In: Proc. Programming Languages: Implementations, Logics, and Programs
Coen ES, Meyerowitz EM (1991) The war of the whorls: genetic interactions controlling flower development. Nature pp 31–37
Corblin F, Tripodi S, Fanchon E, Ropers D, Trilling L (2009) A declarative constraint-based method for analysing discrete genetic regulatory networks. Biosystems 98:91–104
Corblin F, Fanchon E, Trilling L (2010) Applications of a formal approach to decipher discrete genetic networks. BMC Bioinformatics 11:385
Corblin F, Bordeaux F, Fanchon E, Hamadi Y, Trilling L (2011) Connections and integration with sat solvers: A survey and a case study in computational biology. In: Springer (ed) Hybrid Optimization: Optimization and Its Applications, vol 45, pp 425–461
Demongeot J, Aracena J, Thuderoz F, Baum TP, Cohen O (2003) Genetic regulation networks: circuits, regulons and attractors. C R Biologies 326:171–188
Demongeot J, Elena A, Sené S (2008) Robustness in neural and genetic networks. Acta Biotheor 56:27–49
Dubrova E, Teslenko M (2011) A sat-based algorithm for finding attractors in synchronous boolean networks. IEEE/ACM Trans Comp Biol Bioinfo 8:1393–1399
Eén N, Biere A (2005) Effective preprocessing in SAT through variable and clause elimination. In: SAT'2005 – Theory and Applications of Satisfiability Testing, LNCS 3569
Eén N, Sörensson N (2004) An extensible SAT-solver. In: SAT'2003 – Theory and Applications of Satisfiability Testing, LNCS 2919
Elena A (2009) Robustesse des réseaux d'automates booléens à seuil aux modes d'itération. application à la modélisation des réseaux de régulation génétique. PhD thesis, Université Joseph Fourier, Grenoble
Giacomantonio EC, Goodhill GJ (2010) A boolean model of the gene regulatory network underlying mammalian cortical area development. PLoS Comput Biol 6:e1000,936, DOI 10.1371/journal.pcbi.1000936
Glade N, Elena A, Corblin F, Fanchon E, Demongeot J, Ben-Amor H (2011) Determination, optimization and taxonomy of regulatory networks. the example of *Arabidopsis thaliana* flower morphogenesis. In: IEEE (ed) IEEE Proceedings of International Conference on Advanced Information Networking and Applications Workshops, AINA' 11 and BLSMC' 11, Singapore, IEEE Proceedings, Psicataway
Gowda T, Vrudhula S, Seungchan K (2009) Prediction of pairwise gene interaction using threshold logic. The challenges of systems biology: Annals of the New York Academy of Sciences 1158(1):276–286
Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci USA 79:2554–2558

Kauffman S (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. J Theor Biol 22:437–467

Mendoza L, Alvarez-Buylla E (1998) Dynamics of the genetic regulatory network: Arabidopsis thaliana flower morphogenesis. Journal of Theoretical Biology 193(2):307–319

Remy E, Ruet P, Thieffry D (2008) Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. Adv Appl Math 41:335–350

Richard A (2010) Negative circuits and sustained oscillations in asynchronous automata networks. Adv Appl Math 44:378–392

Thomas R (1980) On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations. Springer series in synergetics 9:180–193